

# An iterative method to compute the sign function of a non-Hermitian matrix and its application to the overlap Dirac operator at nonzero chemical potential

J. Bloch<sup>a</sup>, A. Frommer<sup>b</sup>, B. Lang<sup>b</sup>, and T. Wettig<sup>a</sup>

<sup>a</sup>*Institute for Theoretical Physics, University of Regensburg, 93040 Regensburg, Germany*

<sup>b</sup>*Department of Mathematics, University of Wuppertal, 42097 Wuppertal, Germany*

---

## Abstract

The overlap Dirac operator in lattice QCD requires the computation of the sign function of a matrix. While this matrix is usually Hermitian, it becomes non-Hermitian in the presence of a quark chemical potential. We show how the action of the sign function of a non-Hermitian matrix on an arbitrary vector can be computed efficiently on large lattices by an iterative method. A Krylov subspace approximation based on the Arnoldi algorithm is described for the evaluation of a generic matrix function. The efficiency of the method is spoiled when the matrix has eigenvalues close to a function discontinuity. This is cured by adding a small number of critical eigenvectors to the Krylov subspace, for which we propose two different deflation schemes. The ensuing modified Arnoldi method is then applied to the sign function, which has a discontinuity along the imaginary axis. The numerical results clearly show the improved efficiency of the method. Our modification is particularly effective when the action of the sign function of the same matrix has to be computed many times on different vectors, e.g., if the overlap Dirac operator is inverted using an iterative method.

*Key words:* overlap Dirac operator, quark chemical potential, sign function, non-Hermitian matrix, iterative methods

*PACS:* 02.60.Dc, 11.15.Ha, 12.38.Gc

---

## 1. Introduction

The only systematic nonperturbative approach to quantum chromodynamics (QCD) is the numerical simulation of the theory on a finite space-time lattice. For a long time, the implementation of chiral symmetry on the lattice posed serious problems [1], but these problems have recently been solved in a number of complementary ways. Perhaps the most prominent solution is the overlap Dirac operator [2] which provides an exact solution of the Ginsparg-Wilson relation [3]. However, the price one has to pay for this solution is the numerical computation of the sign function of a sparse matrix  $A$  of dimension  $N$ . Here and in the following, *computing some function  $f$  of a matrix  $A$*  is a short-hand for *computing  $f(A) \cdot x$* , where  $x \in \mathbb{C}^N$ , i.e., determining the action of  $f(A)$  on the vector  $x$ . Typically  $A$  is Hermitian, and efficient methods to compute its sign function have been developed for this case [4,5].

The phase diagram of QCD is currently being explored experimentally in relativistic heavy ion collisions and theoretically in lattice simulations and model calculations [6]. To describe QCD at nonzero density, a quark chemical potential is introduced in the QCD Lagrangian. If this chem-

ical potential is implemented in the overlap operator [7], the matrix  $A$  loses its Hermiticity, and one is faced with the problem of computing the sign function of a non-Hermitian sparse matrix. On a small lattice, this can be done by performing a full diagonalization and using the spectral matrix function definition (see Eq. (4) below), but on larger lattices one needs to resort to iterative methods to keep the computation time and memory requirements under control.

The purpose of this paper is to introduce such an iterative method. In the next section we describe the non-Hermitian problem in more detail and briefly discuss the sign function for non-Hermitian matrices. In Sec. 4 we propose an Arnoldi-based method to make a Krylov subspace approximation of a generic matrix function. The efficiency of this method is poor when computing the sign function of a matrix having eigenvalues with small absolute real parts. This is caused by the discontinuity of the sign function along the imaginary axis. In Sec. 5 we enhance the Arnoldi method by taking into account exact information about these critical eigenvalues. We use the method to compute the sign function occurring in the overlap Dirac operator of lattice QCD, and in Sec. 6 we discuss the results obtained for two different lattice sizes.

## 2. The overlap operator and the sign function

The overlap formulation of the Dirac operator is a rigorous method to preserve chiral symmetry at finite lattice spacing in a vector-like gauge theory. Its construction is based on successive developments described in seminal papers by Kaplan, Shamir, Furman, Narayanan and Neuberger [8–10,2]. In the presence of a non-zero quark chemical potential  $\mu$ , the massless overlap Dirac operator is given by [7]

$$D_{\text{ov}}(\mu) = 1 + \gamma_5 \text{sgn}(H_w(\mu)), \quad (1)$$

where  $\text{sgn}$  stands for the sign function,  $H_w(\mu) = \gamma_5 D_w(\mu)$ ,  $D_w(\mu)$  is the Wilson-Dirac operator at nonzero chemical potential [11,12] with negative Wilson mass  $m_w \in (-2, 0)$ ,  $\gamma_5 = \gamma_1 \gamma_2 \gamma_3 \gamma_4$ , and  $\gamma_\nu$  with  $\nu = 1, \dots, 4$  are the Dirac gamma matrices in Euclidean space. The Wilson-Dirac operator is a discretized version of the continuum Dirac operator that avoids the replication of fermion species which occurs when a naive discretization of the derivative operator is used. It is given by

$$\begin{aligned} [D_w(\mu)]_{nm} &= \delta_{n,m} \\ &- \kappa \sum_{j=1}^3 (1 + \gamma_j) U_{n,j} \delta_{n+\hat{j},m} - \kappa \sum_{j=1}^3 (1 - \gamma_j) U_{n-\hat{j},j}^\dagger \delta_{n-\hat{j},m} \\ &- \kappa (1 + \gamma_4) e^\mu U_{n,4} \delta_{n+\hat{4},m} - \kappa (1 - \gamma_4) e^{-\mu} U_{n-\hat{4},4}^\dagger \delta_{n-\hat{4},m}, \end{aligned} \quad (2)$$

where  $\kappa = 1/(8 + 2m_w)$  and  $U_{n,\nu}$  is the  $SU(3)$ -matrix associated with the link connecting the lattice site  $n$  to  $n + \hat{\nu}$ . The exponential factors  $e^{\pm\mu}$  are responsible for the non-Hermiticity of the operator. The quark field at each lattice site corresponds to 12 variables: 3  $SU(3)$  color components  $\times$  4 Dirac spinor components.

For  $\mu \neq 0$  the argument  $H_w(\mu)$  of the sign function becomes non-Hermitian, and we need to define the sign function for this case. Consider first a given matrix  $A$  with no particular symmetry properties and a function  $f$ . Let  $\Gamma$  be a collection of closed contours in  $\mathbb{C}$  such that  $f$  is analytic inside and on  $\Gamma$  and such that  $\Gamma$  encloses the spectrum of  $A$ . Then the function  $f(A)$  of the matrix  $A$  can be defined by [13]

$$f(A) = \frac{1}{2\pi i} \oint_{\Gamma} f(z) (zI - A)^{-1} dz, \quad (3)$$

where the integral is defined component-wise and  $I$  denotes the identity matrix.

From this definition it is easy to derive a spectral function definition, even if the matrix is non-Hermitian. If the matrix  $A$  is diagonalizable, i.e.,  $A = U \Lambda U^{-1}$  with a diagonal eigenvalue matrix  $\Lambda = \text{diag}(\lambda_i)$  and  $U \in \text{Gl}(N, \mathbb{C})$ , then

$$f(A) = U f(\Lambda) U^{-1} \quad (4)$$

with

$$f(\Lambda) = \text{diag}(f(\lambda_i)). \quad (5)$$

If  $A$  cannot be diagonalized, a more general spectral definition of  $f(A)$  can be derived from Eq. (3) using the Jordan decomposition  $A = U(\bigoplus_i J_i)U^{-1}$  with Jordan blocks

$$J_i = \begin{pmatrix} \lambda_i & 1 & \cdots & 0 \\ 0 & \lambda_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & \lambda_i \end{pmatrix}. \quad (6)$$

Then,

$$f(A) = U \left( \bigoplus_i f(J_i) \right) U^{-1}, \quad (7)$$

where

$$f(J_i) = \begin{pmatrix} f(\lambda_i) & f^{(1)}(\lambda_i) & \cdots & \frac{f^{(m_i-1)}(\lambda_i)}{(m_i-1)!} \\ 0 & f(\lambda_i) & \ddots & \vdots \\ \vdots & \ddots & \ddots & f^{(1)}(\lambda_i) \\ 0 & \cdots & 0 & f(\lambda_i) \end{pmatrix} \quad (8)$$

with  $m_i$  the size of the Jordan block, see [14]. The superscripts denote derivatives of the function with respect to its argument.

Non-Hermitian matrices typically have complex eigenvalues, and applying Eq. (4) or (7) to the sign function in Eq. (1) requires the evaluation of the sign of a complex number. The sign function needs to satisfy  $[\text{sgn}(z)]^2 = 1$  and, for real  $x$ ,  $\text{sgn}(x) = \pm 1$  if  $x \gtrless 0$ . These properties are satisfied if one defines

$$\text{sgn}(z) \equiv \frac{z}{\sqrt{z^2}} = \text{sgn}(\text{Re}(z)), \quad (9)$$

where in the last equality the cut of the square root is chosen along the negative real axis. Using the definition (9) in the spectral definition (7) yields a matrix sign function in agreement with that used in [15–18]. Indeed, based on the general Jordan decomposition

$$A = U \begin{pmatrix} J_+ & \\ & J_- \end{pmatrix} U^{-1}, \quad (10)$$

where  $J_+$  represents the Jordan blocks corresponding to the eigenvalues with positive real part and  $J_-$  those corresponding to the eigenvalues with negative real part, the spectral definition (7) for the sign function becomes

$$\text{sgn}(A) = U \begin{pmatrix} +I & \\ & -I \end{pmatrix} U^{-1}, \quad (11)$$

see also [19]. This definition agrees with the result one obtains when deriving Eq. (1) from the domain-wall fermion formalism at  $\mu \neq 0$  in the limit in which the extent of the fifth dimension goes to infinity [20].

For any square matrix  $A$  we have  $\text{sgn}(A)^2 = I$ , and a short calculation [7] shows that for this reason the overlap operator  $D_{\text{ov}}(\mu)$  as defined in Eq. (1) satisfies the Ginsparg-Wilson relation

$$\{D_{\text{ov}}, \gamma_5\} = D_{\text{ov}} \gamma_5 D_{\text{ov}}. \quad (12)$$

If  $A$  is Hermitian, the polar factor  $\text{pol}(A) = A(A^\dagger A)^{-1/2}$  of  $A$  coincides with  $\text{sgn}(A)$ , and this fact has been used successfully to develop efficient iterative methods for computing the action of the matrix sign function on a vector [21]. However, if  $A$  is non-Hermitian, then in general  $\text{sgn}(A) \neq \text{pol}(A)$  and  $\text{pol}(A)^2 \neq I$ . Thus, for  $\mu \neq 0$ , replacing  $\text{sgn}(H_w)$  by  $\text{pol}(H_w)$  in the definition of the overlap operator in Eq. (1) not only changes the operator but also violates the Ginsparg-Wilson relation, as can also be seen in numerical experiments. We conclude that the definition given in Eq. (1) is the correct formulation of the overlap operator for  $\mu \neq 0$ .

### 3. Direct and iterative methods

A numerical implementation of the sign function using the spectral definition (4) is only possible for small matrices, as a full diagonalization becomes too expensive as the matrix grows. As an alternative, matrix-based iterative algorithms for the computation of the matrix sign function have been around for many years [15,17,22,23]. Although these algorithms are much faster than the direct implementation of the spectral definition for medium-sized problems, they still require the storage and manipulation (i.e., inversions and/or multiplications) of the entire matrix. This is feasible for medium-sized matrices, but becomes too expensive for the very large matrices occurring in typical lattice QCD simulations. E.g., even for an  $8^3 \times 8$  lattice, which is the minimum lattice size required for a physically relevant problem, the matrix dimension is already  $12 \cdot 8^3 \cdot 8 \approx 50\,000$ . Even though these QCD matrices are sparse, the iterative procedure computing the sign function fills the matrix as the iterations proceed, and the method eventually becomes prohibitively expensive.

Therefore, another iterative method is required which does not produce an approximation to the full sign matrix itself, but rather produces an approximation to the vector  $y = \text{sgn}(A)x$ , i.e., to the operation of the sign matrix on an arbitrary vector. Many QCD applications only require the knowledge of this product for a number of selected source vectors  $x$ . For instance, some low-energy properties of QCD can be described by the lowest-lying eigenvalues of the Dirac operator. These eigenvalues can efficiently be found by an iterative eigenvalue solver like ARPACK [24], which only requires the computation of matrix-vector multiplications. Analogously, the computation of the propagation of fermions can be well approximated by inverting the Dirac operator on a selected number of source vectors  $b_k$ , i.e., the solution of the systems  $D_{\text{ov}}x = b_k$ . These inversions are also performed using iterative linear solvers requiring only matrix-vector multiplications.

Such iterative methods, mostly from the class of Krylov subspace methods, are already extensively used for the solution of eigenvalue problems, linear systems, and for function evaluations [25,26] with Hermitian matrices. There, the ancestor of all methods is the Lanczos method, of which

many variants and improvements have been built over the years. The Lanczos method makes use of short recurrences to build an orthonormal basis in the Krylov subspace.

Krylov subspace methods are also used for non-Hermitian matrices in the context of eigenvalue problems (a popular example being the restarted Arnoldi method of ARPACK), for the solution of linear systems, and even for the evaluation of the exponential function [27,28]. The two most widely used methods are the Arnoldi method and the two-sided Lanczos method. In contrast to the Hermitian case, the Arnoldi method requires long recurrences to construct an orthonormal basis for the Krylov subspace, while the two-sided Lanczos method uses two short recurrence relations, but at the cost of losing orthogonality.

In the next section we present an Arnoldi-based method to compute a generic function of a non-Hermitian matrix. The application of the two-sided Lanczos method to this problem will be investigated in a separate publication.

### 4. Arnoldi function approximation for a non-Hermitian matrix

From the spectral definition (4) it follows that  $f(A)$  is identical to some polynomial  $P_{K-1}(A)$  of degree  $K-1 < N$ . Indeed, the unique interpolation polynomial  $P_{K-1}(z)$ , which interpolates  $f(z)$  at the different eigenvalues  $\lambda_i$  of  $A$ , satisfies  $f(A) = P_{K-1}(A)$ , as follows immediately from Eq. (4). If  $A$  has non-trivial Jordan blocks, this is still true, but interpolation is now in the Hermite sense, where at each eigenvalue  $P_{K-1}$  interpolates  $f$  and its derivatives up to order one less than the size of the largest corresponding Jordan block, see [19].

Hence, for an arbitrary vector  $x$ ,

$$y \equiv f(A)x = P_{K-1}(A)x = \sum_{i=0}^{K-1} c_i A^i x. \quad (13)$$

The idea is to construct an approximation to  $y$  using a polynomial of degree  $k-1$  with  $k \ll N$ . One possibility is to construct a *good* polynomial approximation  $P_{k-1}(A)$  such that  $P_{k-1}(\lambda_i) \approx f(\lambda_i)$ . This would yield a single polynomial approximation operating on any vector  $x$  to compute  $f(A)x \approx P_{k-1}(A)x$ . One such example is the expansion in terms of Chebyshev polynomials up to degree  $k-1$ . Although Chebyshev polynomials are very close to the minimax polynomial for a function approximation over an appropriate ellipse in the complex plane, one can do better for matrix function approximations. First of all, one only needs a good approximation to  $f$  at the eigenvalues of  $A$ , and secondly, one can use information about the source vector  $x$  to improve the polynomial approximation. The vector  $x$  can be decomposed using the (generalized) eigenvectors of  $A$  as a complete basis, and clearly some eigendirections will be more relevant than others in the function approximation. Using a fixed interpolation polynomial does not use any information about the vector  $x$ . Furthermore, the only

feature of  $A$  usually taken into account by such polynomial approximations is the extent of its spectrum.

Indeed, there exists a *best* polynomial approximation  $\hat{y} = P_{k-1}(A)x$  of degree at most  $k-1$ , which is readily defined as the orthogonal projection of  $y$  on the Krylov subspace  $\mathcal{K}_k(A, x) = \text{span}(x, Ax, A^2x, \dots, A^{k-1}x)$ . If  $V_k = (v_1, \dots, v_k)$  is an  $N \times k$  matrix whose columns form an orthonormal basis in  $\mathcal{K}_k$ , then  $V_k V_k^\dagger$  is a projector on the Krylov space, and the projection is  $\hat{y} = V_k V_k^\dagger y$ .

The operation of any polynomial of  $A$  of degree smaller than  $k$  on  $x$  is also an element of  $\mathcal{K}_k$ , and the projection  $\hat{y}$  corresponds to the polynomial which minimizes  $\Delta = f(A)x - P_{k-1}(A)x$ , as  $\Delta \perp \mathcal{K}_k(A, x)$ . Clearly, this approximation explicitly takes into account information about  $A$  and  $x$ .

The problem, however, is that to find this *best* polynomial approximation of degree  $k-1$  one already has to know the answer  $y = f(A)x$ . Therefore, we need a method to approximate the projected vector  $\hat{y}$ . This can be done by one of the Krylov subspace methods mentioned in Sec. 3. Here, we use the Arnoldi algorithm to construct an orthonormal basis for the Krylov subspace  $\mathcal{K}_k(A, x)$  using the long recurrence

$$AV_k = V_k H_k + \beta_k v_{k+1} e_k^T, \quad (14)$$

where  $v_1 = x/\beta$ ,  $\beta = |x|$ ,  $H_k$  is an upper Hessenberg matrix (upper triangular + one subdiagonal),  $\beta_k = H_{k+1,k}$ , and  $e_k$  is the  $k$ -th basis vector in  $\mathbb{C}^k$ . The projection  $\hat{y}$  of  $f(A)x$  on  $\mathcal{K}_k(A, x)$  can be written as

$$\hat{y} = V_k V_k^\dagger f(A)x. \quad (15)$$

Making use of  $x = \beta v_1 = \beta V_k e_1$ , Eq. (15) becomes

$$\hat{y} = \beta V_k V_k^\dagger f(A) V_k e_1. \quad (16)$$

From Eq. (14) it is easy to see that

$$H_k = V_k^\dagger A V_k \quad (17)$$

as  $V_k^\dagger V_k = I_k$  and  $v_{k+1} \perp V_k$ . Therefore it seems natural to introduce the approximation [27]

$$f(H_k) \approx V_k^\dagger f(A) V_k \quad (18)$$

in Eq. (16), which finally yields

$$\hat{y} \approx \beta V_k f(H_k) e_1. \quad (19)$$

This expression is just a linear combination of the  $k$  Arnoldi vectors  $v_i$ , where the  $k$  coefficients are given by  $\beta$  times the first column of  $f(H_k)$ . Saad [29] showed that the approximation (19) corresponds to replacing the polynomial interpolating  $f$  at the eigenvalues of  $A$  by the lower-degree polynomial which interpolates  $f$  at the eigenvalues of  $H_k$ , which are also called the Ritz values of  $A$  in the Krylov space. The hope is that for  $k$  not too large the approximation (19) will be a suitable approximation for  $y$ . The approximation of  $f(A)x$  by Eq. (19) replaces the computation of  $f(A)$  by that of  $f(H_k)$ , where  $H_k$  is of much smaller size than  $A$ . The evaluation of (the first column of)  $f(H_k)$  can be implemented using a spectral decomposition, or another suitable evaluation method [15,17,22,23].

The long recurrences of the Arnoldi method make the method much slower than the Lanczos method used for Hermitian matrices. Nevertheless, our first results showed consistent convergence properties when computing the matrix sign function: as the size of the Krylov space increases, the method converges to within machine accuracy. More precisely, for the sign function the method shows a see-saw profile corresponding to even-odd numbers of Krylov vectors, as was previously noticed for the Hermitian case as well [5]. This even-odd pattern is related to the sign function being odd in its argument. The see-saw effect is completely removed when using the Harmonic Ritz values as described in Ref. [5] for Hermitian matrices and extended to non-Hermitian matrices in Ref. [28], and convergence becomes smooth. Alternatively, one can just as well restrict the calculations to even-sized Krylov subspaces.

Unfortunately, in the case of the sign function the Arnoldi method described above has a very poor efficiency when some of the eigenvalues are small. In the case considered in Fig. 3 (see Sec. 6 below), the size of the Krylov space has to be taken very large ( $k \approx N/2$ ) to reach accuracies of the order of  $10^{-8}$  (see the  $m = 0$  curve in the top pane of Fig. 3). A discussion and resolution of this problem are given in the next section.

## 5. Deflation

### 5.1. Introduction

For Hermitian matrices, it is well known that the computation of the sign function can be improved by deflating the eigenvalues smallest in absolute value [5]. The reason why this is crucial and specific to the sign function is the discontinuity of the sign function at zero. For non-Hermitian matrices, the situation is analogous since the sign function now has a discontinuity along the imaginary axis. A necessary condition for the method described in Sec. 4 to be efficient is the ability to approximate  $f$  well at the eigenvalues of  $A$  by a low-order polynomial. If the gap between the eigenvalues of  $A$  to the left and to the right of the imaginary axis is small, no low-order polynomial will exist which will be accurate enough for all eigenvalues.

The idea is to resolve this problem by treating these critical eigenvalues exactly, and performing the Krylov subspace approximation on a deflated space.

In the Hermitian case, deflation is straightforward. The function  $f(A)$  of a Hermitian matrix  $A$  with eigenvalues  $\lambda_i$  and orthonormal eigenvectors  $u_i$  ( $i = 1, \dots, N$ ) can be written as

$$f(A) = \sum_{i=1}^N f(\lambda_i) u_i u_i^\dagger, \quad (20)$$

and its operation on an arbitrary vector  $x$  as

$$f(A)x = \sum_{i=1}^N f(\lambda_i) (u_i^\dagger x) u_i. \quad (21)$$



If  $m$  critical eigenvalues of  $A$  and their corresponding eigenvectors have been computed, one can split the vector space into two orthogonal subspaces and write an arbitrary vector as  $x = x_{\parallel} + x_{\perp}$ , where  $x_{\parallel} = \sum_{i=1}^m (u_i^{\dagger} x) u_i$  and  $x_{\perp} = x - x_{\parallel}$ . Eq. (21) can then be rewritten as

$$f(A)x = \sum_{i=1}^m f(\lambda_i) (u_i^{\dagger} x) u_i + f(A)x_{\perp}. \quad (22)$$

The first term on the right-hand side of Eq. (22) can be computed exactly, and the second term can be approximated by applying the Arnoldi method of Sec. 4 to  $x_{\perp}$ . As the vector  $x_{\perp}$  does not contain any contribution in the eigenvector directions corresponding to the critical eigenvalues, the polynomial approximation no longer needs to interpolate  $f$  at the eigenvalues closest to the function discontinuity to approximate  $f(A)x_{\perp}$  well. Therefore, after deflation, lower-degree polynomials will yield the required accuracy, and a smaller-sized Krylov subspace can be used in the approximation. In theory, the orthonormality of the eigenvectors guarantees that the Krylov subspace will be perpendicular to  $x_{\parallel}$ , but in practice numerical inaccuracies could require us to reorthogonalize the subspaces during the construction of the Krylov subspace.

For non-Hermitian matrices the (generalized) eigenvectors are no longer orthonormal, and it is not immediately clear how to deflate a critical subspace. The matrix functions as defined in (4) or (7) involve the inverse of the matrix of basis vectors,  $U$ , and no simple decomposition into orthogonal subspaces can be performed.

In the remainder of this section, we will develop two alternative deflation schemes for the non-Hermitian case, using a composite subspace generated by adding a small number of critical eigenvectors to the Krylov subspace. This idea of an *augmented Krylov subspace method* has been used in the iterative solution of linear systems for some time, see, e.g., Ref. [30]. Since in computational practice one will never encounter non-trivial Jordan blocks, we assume in the following, for simplicity, that the matrix is diagonalizable.

## 5.2. Schur deflation

We construct the subspace  $\Omega_m + \mathcal{K}_k(A, x)$ , which is the sum of the subspace  $\Omega_m$  spanned by the eigenvectors corresponding to  $m$  critical eigenvalues of  $A$  and the Krylov subspace  $\mathcal{K}_k(A, x)$ . The aim is to make an approximation similar to that of Eq. (19), but to treat the contribution of the critical eigenvalues to the sign function explicitly so that the size of the Krylov subspace can be kept small.

Assume that  $m$  critical eigenvalues and right eigenvectors of  $A$  are determined using an iterative eigenvalue solver like the one implemented in ARPACK. From this one can easily construct  $m$  Schur vectors and the corresponding  $m \times m$  upper triangular matrix  $T_m$  satisfying

$$AS_m = S_m T_m, \quad (23)$$

where  $S_m = (s_1, \dots, s_m)$  is the  $N \times m$  matrix formed by the orthonormal Schur vectors and the diagonal elements of  $T_m$  are the eigenvalues corresponding to the Schur vectors. These Schur vectors form an orthonormal basis of the eigenvector subspace  $\Omega_m$ , which is invariant under operation of  $A$ .

After constructing the  $m$ -dimensional subspace  $\Omega_m$  we run a modified Arnoldi method to construct an orthogonal basis of the composite subspace  $\Omega_m + \mathcal{K}_k(A, x)$ . That is, each Arnoldi vector is orthogonalized not only against the previous ones, but also against the Schur vectors  $s_i$ . In analogy to (14), this process can be summarized as

$$A \begin{pmatrix} S_m & V_k \end{pmatrix} = \begin{pmatrix} S_m & V_k \end{pmatrix} \begin{pmatrix} T_m & S_m^{\dagger} A V_k \\ 0 & H_k \end{pmatrix} + \beta_k v_{k+1} e_{m+k}^T. \quad (24)$$

Here, the columns  $v_1, \dots, v_k$  of  $V_k$  form an orthonormal basis of the space  $\mathcal{K}_k^{\perp}(A, x)$ , which is the projection of the Krylov subspace  $\mathcal{K}_k(A, x)$  onto the orthogonal complement  $\Omega^{\perp}$  of  $\Omega_m$ . In particular,  $v_1 = x_{\perp}/\beta$ , where  $x_{\perp} = (1 - S_m S_m^{\dagger})x$  is the projection of  $x$  onto  $\Omega^{\perp}$  and  $\beta = |x_{\perp}|$ . Again,  $H_k$  is an upper Hessenberg matrix.

Note that the orthogonality of  $\mathcal{K}_k^{\perp}$  with respect to  $\Omega_m$  has to be enforced explicitly during the Arnoldi iterations, as the operation of  $A$  on a vector in the projected Krylov subspace  $\mathcal{K}_k^{\perp}$  in general gets a contribution belonging to  $\Omega_m$ , i.e.,  $\mathcal{K}_k^{\perp}$  is not invariant under the operation of  $A$ . This is a consequence of the non-orthogonality of the eigenvectors of  $A$ .

Defining

$$Q = \begin{pmatrix} S_m & V_k \end{pmatrix} \text{ and } H = \begin{pmatrix} T_m & S_m^{\dagger} A V_k \\ 0 & H_k \end{pmatrix}, \quad (25)$$

$H$  satisfies a relation similar to Eq. (17), namely

$$H = Q^{\dagger} A Q, \quad (26)$$

and the function approximation derived in Sec. 4 can be used here as well. We briefly repeat the steps of Sec. 4. The operation of the matrix function  $f(A)$  on the vector  $x$  can be approximated by its projection on the composite subspace,

$$f(A)x \approx Q Q^{\dagger} f(A)x, \quad (27)$$

and because  $x$  lies in the subspace,

$$f(A)x \approx Q Q^{\dagger} f(A) Q Q^{\dagger} x. \quad (28)$$

As  $H$  satisfies Eq. (26), we can introduce the same approximation as in Eq. (18),

$$f(H) \approx Q^{\dagger} f(A) Q, \quad (29)$$

and substituting Eq. (29) in Eq. (28) we construct the function approximation

$$f(A)x \approx Q f(H) Q^{\dagger} x. \quad (30)$$

Because of the block structure (25) of the composite Hessenberg matrix  $H$ , the matrix  $f(H)$  can be written as

$$f(H) = \begin{pmatrix} f(T_m) & Y \\ 0 & f(H_k) \end{pmatrix}. \quad (31)$$

The upper left corner is the function of the triangular Schur matrix  $T_m$  (which is again triangular), and the lower right corner is the (dense) matrix function of the Arnoldi Hessenberg matrix  $H_k$ . The upper right corner reflects the coupling between both subspaces and is given by the solution of the Sylvester equation

$$T_m Y - Y H_k = f(T_m) X - X f(H_k) \quad (32)$$

with  $X = S_m^\dagger A V_k$ , which follows from the identity  $f(H)H = Hf(H)$ . Combining (30) and (31), we obtain

$$\begin{aligned} f(A)x &\approx \begin{pmatrix} S_m & V_k \end{pmatrix} \begin{pmatrix} f(T_m) & Y \\ 0 & f(H_k) \end{pmatrix} \begin{pmatrix} S_m^\dagger \\ V_k^\dagger \end{pmatrix} x \\ &= S_m f(T_m) S_m^\dagger x + \begin{pmatrix} S_m & V_k \end{pmatrix} \begin{pmatrix} Y \\ f(H_k) \end{pmatrix} V_k^\dagger x. \end{aligned} \quad (33)$$

Note that  $V_k^\dagger x = \beta e_1$ . Therefore only  $f(T_m)$  and the first column of  $Y$  and  $f(H_k)$ , i.e., the first  $m+1$  columns of  $f(H)$ , are needed to evaluate (33). This information can be computed using the spectral definition (4) or some other suitable method [15,17,22,23]. In the case of the sign function we chose to use Roberts' iterative method [16]

$$S^{n+1} = \frac{1}{2} [S^n + (S^n)^{-1}] \quad (34)$$

with  $S^0 = A$ , which converges quadratically to  $\text{sgn}(A)$ . Roberts' method is applied to compute  $\text{sgn}(T_m)$  and  $\text{sgn}(H_k)$ . The matrix  $Y$  is computed by solving the Sylvester equation (32) using the method described in Appendix A.

In the implementation one has to be careful to compute the deflated eigenvectors to high enough accuracy, as this will limit the overall accuracy of the function approximation. When computing  $f(A)x$  for several  $x$ , the partial Schur decomposition (23) and the triangular matrix  $f(T_m)$  need to be computed only once. Only the modified Arnoldi method must be repeated for each new vector  $x$ .

We summarize our algorithm for approximating  $f(A)x$ :

- (i) Determine the eigenvectors for  $m$  critical eigenvalues of  $A$  using ARPACK. Construct and store the corresponding Schur matrix  $S_m$  and the upper triangular matrix  $T_m = S_m^\dagger A S_m$ . The columns of  $S_m$  form an orthonormal basis of a subspace  $\Omega_m$ .
- (ii) Compute the triangular matrix  $f(T_m)$  using (34).
- (iii) Compute  $x_\perp = (1 - S_m S_m^\dagger)x$ .
- (iv) Construct an orthonormal basis for the projected Krylov subspace  $\mathcal{K}_k^\perp(A, x_\perp)$  using a modified Arnoldi method. The basis is constructed iteratively by orthogonalizing each new Krylov vector with respect to  $\Omega_m$  and to all previous Arnoldi vectors, and is stored as columns of a matrix  $V_k$ . Also build the upper Hessenberg matrix  $H = Q^\dagger A Q$  with  $Q = (S_m, V_k)$ .

- (v) Compute (column  $m+1$  of)  $f(H)$  using Roberts' iterative method (34) on  $H_k$  and solving the Sylvester equation (32) for  $Y$  as described in Appendix A.
- (vi) Compute the approximation  $f(A)x \approx Q f(H) Q^\dagger x$  using formula (33).

If  $f(A)x$  has to be computed for several  $x$ , only steps (iii)-(vi) need to be repeated for each vector  $x$ .

### 5.3. LR-deflation

An alternative deflation in the same composite subspace  $\Omega_m + \mathcal{K}_k(A, x)$  can be constructed using both the left and right eigenvectors corresponding to the critical eigenvalues. This deflation algorithm is a natural extension of the method described in Sec. 5.1 from the Hermitian to the non-Hermitian case. A similar idea has been used for the iterative solution of linear systems [31,32].

Assume that  $m$  critical eigenvalues of  $A$  have been computed together with their corresponding left and right eigenvectors by some iterative method like the one provided by ARPACK. The right eigenvectors satisfy

$$A R_m = R_m \Lambda_m \quad (35)$$

with  $\Lambda_m$  the diagonal eigenvalue matrix for the  $m$  critical eigenvalues and  $R_m = (r_1, \dots, r_m)$  the matrix of right eigenvectors (stored as columns). Similarly, the left eigenvectors obey

$$L_m^\dagger A = \Lambda_m L_m^\dagger, \quad (36)$$

where  $L_m = (\ell_1, \dots, \ell_m)$  is the matrix containing the left eigenvectors (also stored as columns). For a non-Hermitian matrix, the left and right eigenvectors corresponding to different eigenvalues are orthogonal (for degenerate eigenvalues linear combinations of the eigenvectors can be formed such that this orthogonality property remains valid in general). Furthermore, if the eigenvectors are normalized such that  $\ell_i^\dagger r_i = 1$ , then clearly  $L_m^\dagger R_m = I_m$ , and  $R_m L_m^\dagger$  is an oblique projector on the subspace  $\Omega_m$  spanned by the right eigenvectors.

Let us now decompose  $x$  as

$$x = x_\parallel + x_\perp, \quad (37)$$

where  $x_\parallel = R_m L_m^\dagger x$  is an oblique projection of  $x$  on  $\Omega_m$  and  $x_\perp = x - x_\parallel$ .

Applying  $f(A)$  to the decomposition (37) yields

$$f(A)x = f(A)R_m L_m^\dagger x + f(A)x_\perp. \quad (38)$$

The first term on the right-hand side can be evaluated exactly using

$$f(A)R_m L_m^\dagger x = R_m f(\Lambda_m) L_m^\dagger x, \quad (39)$$

which follows from Eq. (35), while the second term can be approximated by applying the Arnoldi method described in Sec. 4 to the vector  $x_\perp$ . An orthonormal basis is constructed in the Krylov subspace  $\mathcal{K}_k(A, x_\perp)$  using the recurrence

$$A V_k = V_k H_k + \beta_k v_{k+1} e_k^T, \quad (40)$$

where  $v_1 = x_\ominus/\beta$  and  $\beta = |x_\ominus|$ . By construction,  $x_\ominus$  has no components along the  $m$  critical (right) eigendirections, and successive operations of  $A$  will yield no contributions along these directions either, hence  $\mathcal{K}_k(A, x_\ominus)$  does not mix with  $\Omega_m$ . In principle, numerical inaccuracies accumulated during the Arnoldi iterations might make it necessary to occasionally re-extract spurious components along the critical eigendirections. However, this turned out not to be necessary in our numerical calculations.

Applying the Arnoldi approximation (19) to Eq. (38) yields the final approximation

$$f(A)x \approx R_m f(\Lambda_m) L_m^\dagger x + \beta V_k f(H_k) e_1. \quad (41)$$

Note that again only the first column of  $f(H_k)$  is needed to evaluate Eq. (41). As before,  $f(H_k)$  has to be computed using some suitable method. The function  $\text{sgn}(H_k)$  can be efficiently computed using Roberts' algorithm (34).

We summarize our algorithm for approximating  $f(A)x$  in the LR-deflation scheme:

- (i) Determine the left and right eigenvectors for  $m$  critical eigenvalues of  $A$  using ARPACK. Store the corresponding eigenvector matrices  $L_m$  and  $R_m$ .
- (ii) Compute  $f(\lambda_i)$  ( $i = 1, \dots, m$ ) for the critical eigenvalues.
- (iii) Compute  $x_\ominus = (1 - R_m L_m^\dagger)x$ .
- (iv) Construct an orthonormal basis for the Krylov subspace  $\mathcal{K}_k(A, x_\ominus)$  using the Arnoldi recurrence. The basis is constructed iteratively by orthogonalizing each new Krylov vector with respect to all previous Arnoldi vectors, and is stored as columns of a matrix  $V_k$ . Also build the upper Hessenberg matrix  $H_k = V_k^\dagger A V_k$ .
- (v) Compute (the first column of)  $f(H_k)$  using Roberts' iterative method (34).
- (vi) Compute the approximation to  $f(A)x$  using Eq. (41). If  $f(A)x$  has to be computed for several  $x$ , only steps (iii)-(vi) need to be repeated for each vector  $x$ .

#### 5.4. Discussion

We briefly compare both deflation schemes. Although both schemes use the same composite subspace  $\Omega_m + \mathcal{K}_k(A, x)$ , they yield different function approximations resulting from a different subspace decomposition.

In the Schur deflation, the composite subspace is decomposed in a sum of two *orthogonal* subspaces which are coupled by  $A$ , while in the LR-deflation the subspaces no longer mix, at the expense of losing orthogonality of the two subspaces.

Accordingly, both schemes introduce different approximations to  $f(A)x$ : the Schur deflation approximates the orthogonal projection of the solution vector on the total composite space using Eq. (29), while the LR-deflation first extracts the critical eigendirections and only approximates the orthogonal projection of the residual vector on the Krylov subspace using Eq. (19). Therefore, in the Schur deflation the components of  $f(A)x$  along the Schur vectors become more accurate as the Krylov subspace becomes

larger, while in the LR-deflation the components of  $f(A)x$  along the critical eigendirections can be computed exactly, independently of the size of the Krylov subspace. This is probably the reason for the observation that, for fixed  $m$  and  $k$ , the LR-deflation is slightly more accurate than the Schur deflation, see Fig. 4 below.

Numerically, the LR-deflation has two advantages over the Schur deflation. First, its Arnoldi method does not require the deflated directions to be (obliquely) projected out of the Krylov subspace because the subspaces do not mix. Second, this absence of coupling between the subspaces means that the LR-scheme has no analog of the Sylvester equation (32).

A downside of the LR-deflation is that both left and right eigenvectors need to be computed in the initialization phase, whereas the Schur deflation only needs the right eigenvectors. Hence, the Schur deflation will have a shorter initialization phase, unless one needs to operate with both  $f(A)$  and its adjoint  $f(A)^\dagger$ , in which case both sets of eigenvectors are needed anyways (for the latter, the roles of left and right eigenvectors are interchanged).

In the next section, we will present numerical results obtained with our modified Arnoldi method.

## 6. Results

We implemented the modified Arnoldi method proposed in the previous section to compute the sign function occurring in the overlap Dirac operator (1) of lattice QCD.

First we discuss the critical eigenvalues of the  $\gamma_5$ -Wilson-Dirac operator  $H_w(\mu)$ , which are needed for deflation and have to be computed once for any given  $SU(3)$  gauge configuration. Deflation is needed because of the existence of eigenvalues close to the imaginary axis. In Fig. 1 we show the spectrum of  $H_w(\mu)$  for a  $4^4$  lattice and a  $6^4$  lattice, using the same parameters as in Ref. [7], i.e.,  $m_w = -2$  and gauge coupling  $\beta_g = 5.1$ . These complete spectra were computed with LAPACK [33]. This is a very costly calculation, especially for the  $6^4$  lattice, which was done for the sole purpose of this numerical investigation but cannot be performed routinely during production runs.

Although the eigenvalues of interest for deflation in the case of the sign function are those with smallest absolute real parts, we decided to deflate the eigenvalues with smallest magnitude instead. Numerically the latter are more easily determined, and both choices yield almost identical deflations for the  $\gamma_5$ -Wilson operator at nonzero chemical potential. The reason for this is that, as long as the chemical potential does not grow too large, the spectrum looks like a very narrow bow-tie shaped strip along the real axis (see Fig. 1), and the sets of eigenvalues with smallest absolute real parts and smallest magnitudes will coincide.

In practice we compute the eigenvalues of  $H_w$  with smallest magnitude with ARPACK. This package has an option to retrieve the eigenvalues with smallest magnitude without performing an explicit inversion, which would be very ex-

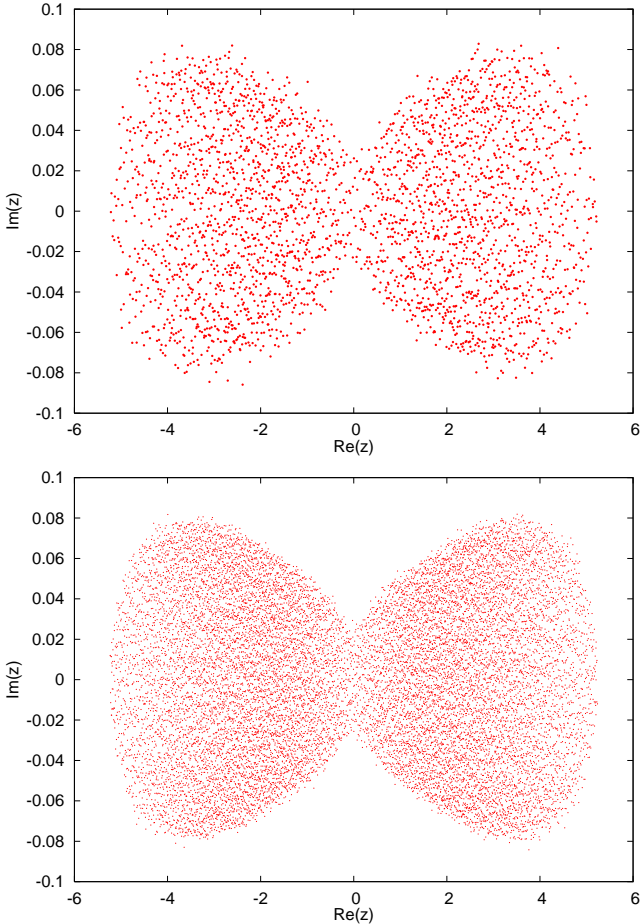


Fig. 1. Spectrum of  $H_w(\mu)$  in Eq. (1) for a  $4^4$  lattice (top pane) and a  $6^4$  lattice (bottom pane), with  $\mu = 0.3$  and  $m_w = -2$ . Note the difference in scale between real and imaginary axes. The gauge fields were generated using the Wilson plaquette action with gauge coupling  $\beta_g = 5.1$  [7].

pensive in this case. However, the use of this option requires the eigenvalues to be near the boundary of the convex hull of the spectrum. From Fig. 1 it is clear that the eigenvalues closest to the origin are deep inside the interior of the convex hull and do not satisfy this requirement. Therefore we opted to compute the eigenvalues with smallest magnitude of the squared operator  $H_w^2$ . Clearly the eigenvalues with smallest magnitude will be the same for both operators, as  $|\lambda^2| = |\lambda|^2$ . The eigenvalues of  $H_w^2$  are given by  $z^2 = x^2 - y^2 + 2ixy$ , where  $x$  and  $y$  are the real and imaginary parts of the eigenvalues  $z$  of  $H_w$ . The spectra of  $H_w^2$  for the  $4^4$  and  $6^4$  lattices are shown in Fig. 2, and clearly the eigenvalues with smallest magnitudes are now close to the boundary of the convex hull of the spectrum so that ARPACK can find them more easily. Since in this approach we square the matrix, there is the potential danger that small eigenvalues get spoiled just by the additional numerical round-off introduced when applying  $A$  twice. However, this should be noticeable only if these eigenvalues are comparable in size to the round-off error. Our calculations are not affected by this problem.

Obviously there is a trade-off between the number of

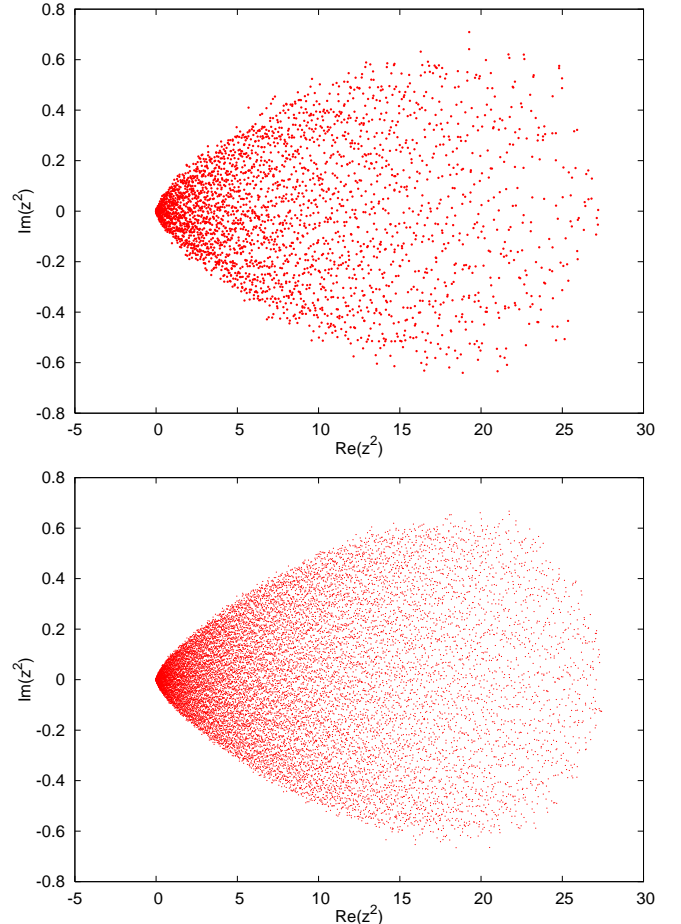


Fig. 2. Spectrum of  $H_w^2(\mu)$  for a  $4^4$  lattice (top pane) and a  $6^4$  lattice (bottom pane), with  $\mu = 0.3$  and  $\beta_g = 5.1$ .

deflated eigenvalues and the size of the Krylov subspace. A useful piece of information in this context is the ratio of the magnitude of the largest deflated eigenvalue over the largest overall eigenvalue, which is given in Table 1 for different numbers of deflated eigenvalues. A comparison of the values for both lattice sizes indicates that the number of eigenvalues with a magnitude smaller than a given ratio increases proportionally with the lattice volume. This is consistent with a spectral density of the small eigenvalues proportional to the lattice volume. This property is also apparent in Figs. 1 and 2, as the contours enclosing the spectra remain unchanged when the volume is increased. As a first guess we therefore expect that scaling  $m$  with the volume should yield comparable convergence properties of the method for various lattice sizes.

The convergence of the method is illustrated in Fig. 3, where the accuracy of the approximation is shown as a function of the Krylov subspace size for two different lattice sizes. The various curves correspond to different numbers of deflated eigenvalues. The results in the figure were computed using the LR-deflation scheme. The Schur deflation yields similar results.

Without deflation ( $m = 0$ ) the Krylov subspace method would be numerically unusable because of the need of large



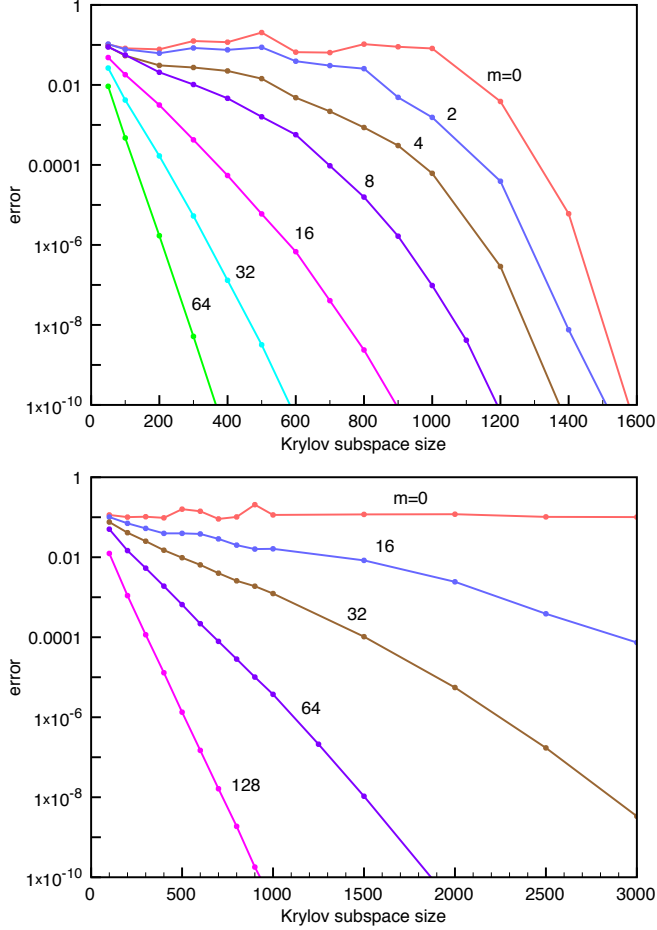


Fig. 3. Accuracy of the modified Arnoldi method for  $y = \text{sgn}(A)x$ . The non-Hermitian matrix is  $A = \gamma_5 D_w(\mu)$ , where  $D_w(\mu)$  is the Wilson-Dirac operator (2) at chemical potential  $\mu = 0.3$ , and  $x = (1, 1, \dots, 1)$ . *Top pane:*  $4^4$  lattice with  $\dim(A) = 3072$ , *bottom pane:*  $6^4$  lattice with  $\dim(A) = 15552$ . The relative error  $\epsilon = \|\tilde{y} - y\|/\|y\|$  is shown as a function of the Krylov subspace size  $k$  for various numbers of deflated eigenvalues  $m$  using the LR-deflation. In order to compute the error  $\epsilon$ , the exact solution  $y$  was first determined using the spectral definition (4) and the full spectral decomposition computed with LAPACK.

Krylov subspaces. Clearly, deflation highly improves the efficiency of the numerical method: as more eigenvalues are deflated, smaller Krylov subspaces are sufficient to achieve

$m$	$\max  \lambda^{\text{defl}}  / \max  \lambda^{\text{all}} $	
	$4^4$ lattice	$6^4$ lattice
2	0.0040	0.0016
4	0.0056	0.0026
8	0.0119	0.0035
16	0.0183	0.0052
32	0.0351	0.0084
64	0.0631	0.0155
128	—	0.0286

Table 1  
Ratio of largest deflated eigenvalue over largest eigenvalue for various numbers of deflated eigenvalues  $m$  for a  $4^4$  and a  $6^4$  lattice.

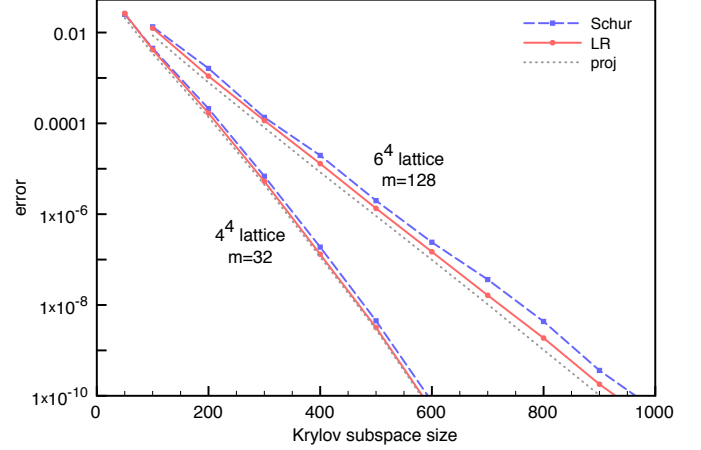


Fig. 4. Comparison of the accuracies achieved with both deflation schemes and the exact projection of  $y$  on the composite space  $\Omega_m + \mathcal{K}_k(A, x)$ . The relative error  $\epsilon = \|\tilde{y} - y\|/\|y\|$  is shown as a function of the Krylov subspace size  $k$ . For both lattice sizes the accuracy of the LR-deflation is slightly better than that of the Schur deflation. Furthermore, the accuracy of the modified Arnoldi approximation is very close to the best possible approximation in the composite subspace.

a given accuracy.

Furthermore, the deflation efficiency grows with increasing lattice volume. To reach an accuracy of  $10^{-8}$  for the  $4^4$  lattice with 25 ( $\approx 0.0081N$ ) deflated eigenvalues, one requires a Krylov subspace size  $k \approx 570$  ( $\approx 0.19N$ ). However, to reach the same accuracy for the  $6^4$  lattice with a comparable deflation of 128 ( $\approx 0.0082N$ ) critical eigenvalues, one only requires  $k \approx 700$  ( $\approx 0.045N$ ). Although the matrix size  $N$  is more than 5 times larger for the  $6^4$  lattice, the Krylov subspace only has to be expanded by a factor of 1.2 to achieve the same accuracy (when  $m$  is scaled proportional to  $N$  so that the ratio of Table 1 remains approximately constant for both lattice sizes).

In Fig. 4 we compare the accuracy of the two deflation schemes described in Sec. 5.2 and 5.3. For an equal number of deflated eigenvalues and equal Krylov subspace size, the LR-deflation seems systematically slightly more accurate than the Schur deflation.

To assess the quality of the modified Arnoldi approximation, it is interesting to compare the approximations (33) and (41) for  $f(A)x$  with the *best approximation* in the composite subspace, which corresponds to the orthogonal projection (27) of  $f(A)x$  on  $\Omega_m + \mathcal{K}_k(A, x)$ ,

$$y_{\text{proj}} = QQ^\dagger y = \sum_{i=1}^m (s_i^\dagger y) s_i + \sum_{i=1}^k (v_i^\dagger y) v_i. \quad (42)$$

The relative accuracy of this projection is also shown in Fig. 4. It is encouraging to note that the modified Arnoldi approximation is quite close to the exact projection  $y_{\text{proj}}$ .

In Table 2 we show the CPU time used by the modified Arnoldi method for the Schur and LR-deflation schemes. The times needed to construct the orthonormal basis in the Krylov subspaces according to Eqs. (24) and (40) and to

compute the sign function of the Arnoldi Hessenberg matrices are tabulated separately. The tabulated times were measured for an  $m = 32$  deflation for the  $4^4$  lattice and  $m = 128$  for the  $6^4$  lattice.

The larger CPU times required by the Schur deflation mainly reflect the additional orthogonalization of the Arnoldi vectors with respect to the Schur vectors. The time

needed to compute the sign of the Hessenberg matrix is also slightly larger for the Schur deflation as it involves the additional solution of the Sylvester Equation (32). For the same reasons, varying  $m$  for a given lattice size will only significantly change the timings for the Schur deflation (this  $m$ -dependence is not shown in the table).

To summarize, the LR-deflation scheme has a somewhat better accuracy and requires less CPU time per iteration than the Schur deflation. The one advantage of the Schur deflation is that it only requires the initial computation of the right eigenvectors, while the LR-deflation requires the computation of both left and right eigenvectors. The time needed to compute the critical eigenvectors of  $H_w(\mu)$  is given in the headers of the four blocks in Table 2. The choice of deflation scheme depends on the number of vectors  $x$  for which  $\text{sgn}(H_w)x$  needs to be computed. This will be the topic of future work on nested iterative methods for non-Hermitian matrices occurring during the inversion of the overlap operator. Of course, as mentioned in Sec. 5.4, if one needs to apply both  $\text{sgn}(H_w)$  and its adjoint, then the LR-deflation will be the better choice.

## 7. Conclusion

In this paper we have proposed an algorithm to approximate the action of a function of a non-Hermitian matrix on an arbitrary vector, when some of the eigenvalues of the matrix lie in a region of the complex plane close to a discontinuity of the function.

The method approximates the solution vector in a composite subspace, i.e., a Krylov subspace augmented by the eigenvectors corresponding to a small number of critical eigenvalues. In this composite subspace two deflation variants are presented based on different subspace decompositions: the Schur deflation uses two coupled orthogonal subspaces, while the LR-deflation uses two decoupled but non-orthogonal subspaces.

The subspace decompositions are then used to compute Arnoldi-based function approximations in which the contribution of the critical eigenvalues is taken into account explicitly. This deflation of critical eigenvalues allows for a smaller size of the Krylov subspace and is crucial for the efficiency of the method.

For the sign function, deflation is particularly important because of its discontinuity along the imaginary axis. The method was applied to the overlap Dirac operator of lattice QCD at nonzero chemical potential, where deflation was shown to clearly enhance the efficiency of the method. If the overlap Dirac operator has to be inverted using some iterative method, each iteration will require the computation of  $\text{sgn}(H_w)x$  for some vector  $x$ . In such a situation the cost for computing the critical eigenvectors, which is done just once, is by far outbalanced by the smaller costs for each evaluation of the sign function. However, an important question that deserves further study is how the optimal number  $m$  of deflated eigenvectors depends on the volume

4 <sup>4</sup> lattice, m = 32 Schur deflation				4 <sup>4</sup> lattice, m = 32 LR-deflation			
initialization time: 14.1 s				initialization time: 27.5 s			
k	Arnoldi	sgn(H)	total	k	Arnoldi	sgn(H <sub>k</sub> )	total
100	0.18	0.03	0.23	100	0.12	0.03	0.15
200	0.59	0.21	0.81	200	0.45	0.20	0.66
300	1.22	0.52	1.75	300	1.01	0.49	1.51
400	2.05	1.08	3.16	400	1.77	1.02	2.82
500	3.12	1.79	4.93	500	2.76	1.69	4.47
600	4.37	2.90	7.31	600	3.94	2.77	6.74
700	5.88	4.57	10.49	700	5.36	4.40	9.79
800	7.56	6.69	14.28	800	6.96	6.44	13.44
900	9.50	9.38	18.92	900	8.84	9.10	17.98
1000	11.63	12.68	24.36	1000	10.84	12.33	23.21

6 <sup>4</sup> lattice, m = 128 Schur deflation				6 <sup>4</sup> lattice, m = 128 LR-deflation			
initialization time: 884 s				initialization time: 1713 s			
k	Arnoldi	sgn(H)	total	k	Arnoldi	sgn(H <sub>k</sub> )	total
100	2.03	0.05	2.13	100	0.66	0.03	0.75
200	5.16	0.22	5.45	200	2.39	0.15	2.62
300	9.27	0.56	9.91	300	5.16	0.42	5.69
400	14.59	1.15	15.85	400	9.01	0.94	10.06
500	20.95	2.09	23.17	500	13.96	1.73	15.84
600	28.12	3.35	31.61	600	20.03	2.80	22.98
700	36.81	5.17	42.15	700	27.09	4.44	31.70
800	46.32	7.39	53.88	800	35.09	6.49	41.78
900	56.83	10.37	67.39	900	44.38	9.10	53.70
1000	68.29	13.88	82.39	1000	54.74	12.36	67.34

Table 2  
CPU time (in seconds) for varying Krylov subspace size  $k$ . *Top row:*  $4^4$  lattice with  $m = 32$ , *bottom row:*  $6^4$  lattice with  $m = 128$ . *Left panes:* Schur deflation, *right panes:* LR-deflation. The time required by the initial calculation of the critical eigenvectors is given in the header of each block. The time needed to construct the Arnoldi basis in the Krylov subspace (column 2) is approximately proportional to  $Nk(k+2m)$  for the Schur deflation and  $Nk^2$  for the LR-deflation. The time used by Roberts' method (34) to compute the sign function of the Hessenberg matrix (column 3) is  $O(k^3)$ . The total time (column 4) also includes the evaluation of Eq. (33) for the Schur deflation and Eq. (41) for the LR deflation. These timings were measured on an Intel Core 2 Duo 2.33GHz computer using optimized ATLAS BLAS routines [34].

and how this influences the initialization time. This question could become performance relevant for large volumes.

As mentioned above, our next steps include the application of the two-sided Lanczos method to the problem of approximating  $f(A)x$  for a non-Hermitian matrix, and the investigation of nested iterative methods for non-Hermitian matrices. Work in these directions is in progress.

## Appendix A. Sylvester equation

In this appendix we describe a particularly simple algorithm to solve the special Sylvester equation

$$TY - YH = C, \quad (\text{A.1})$$

where  $T$  is an  $m \times m$  upper triangular matrix,  $H$  is an  $n \times n$  upper Hessenberg matrix, and the right-hand side  $C$  and the unknown matrix  $Y$  are  $m \times n$  matrices. Classical methods to solve the Sylvester equation when  $T$  and  $H$  are full matrices are formulated in [35,36]. For triangular  $H$  and  $T$  the Sylvester equation can easily be solved by direct substitution, see, e.g., [14]. In principle, this algorithm could also be applied to Eq. (A.1) if the upper Hessenberg matrix  $H$  is first transformed into triangular form using a Schur decomposition. Here we present a more efficient approach, which can be regarded as a natural extension of the algorithm for the triangular Sylvester equation when one of the matrices is upper Hessenberg instead of triangular. Blocking would also be possible (cf., e.g., [37]), but since the solution of the Sylvester equation accounts only for a small portion of the overall time we did not pursue this issue further.

Written out explicitly, the element  $(i, j)$  of the matrix equation (A.1) is

$$\sum_{k=i}^m T_{ik} y_{kj} - \sum_{k=1}^{\max(j+1, n)} y_{ik} H_{kj} = c_{ij} \quad (\text{A.2})$$

for  $i = 1, \dots, m$  and  $j = 1, \dots, n$ .

This matrix equation can be solved row by row from bottom to top, since Eq. (A.2) can be solved for row  $i$  once rows  $i+1, \dots, m$  are known,

$$T_{ii} y_{ij} - \sum_{k=1}^{\max(j+1, n)} y_{ik} H_{kj} = \tilde{c}_{ij} \quad (\text{A.3})$$

with  $\tilde{c}_{ij} = c_{ij} - \sum_{k=i+1}^m T_{ik} y_{kj}$ .

Inside row  $i$  one can solve for the element  $y_{i,j+1}$  as a function of the elements to its left,

$$y_{i,j+1} = -\frac{1}{H_{j+1,j}} \left[ \tilde{c}_{ij} - T_{ii} y_{ij} + \sum_{k=1}^j y_{ik} H_{kj} \right] \quad (\text{A.4})$$

for columns  $j = 1, \dots, n-1$ . From Eq. (A.4) it follows that all elements of row  $i$  can be written as

$$y_{ij} = a_j y_{i1} + b_j, \quad (\text{A.5})$$

where the coefficients  $a_j$  and  $b_j$  can be computed explicitly from the recurrence relations

$$\begin{aligned} a_{j+1} &= -\frac{1}{H_{j+1,j}} \left[ -T_{ii} a_j + \sum_{k=1}^j a_k H_{kj} \right], \\ b_{j+1} &= -\frac{1}{H_{j+1,j}} \left[ \tilde{c}_{ij} - T_{ii} b_j + \sum_{k=1}^j b_k H_{kj} \right], \end{aligned} \quad (\text{A.6})$$

starting from  $a_1 = 1, b_1 = 0$ . After substituting Eq. (A.5) with the known coefficients (A.6), element  $(i, n)$  of Eq. (A.3) can be solved for  $y_{i1}$ ,

$$y_{i1} = \frac{\tilde{c}_{in} - T_{ii} b_n + \sum_{k=1}^n b_k H_{kn}}{T_{ii} a_n - \sum_{k=1}^n a_k H_{kn}}. \quad (\text{A.7})$$

Once  $y_{i1}$  is known, all other elements  $y_{ij}$  of row  $i$  can be computed using Eq. (A.5) with coefficients (A.6).

## Acknowledgments

This work was supported in part by DFG grants FOR465-WE2332/4-2 and Fr755/15-1. JB would like to thank Thomas Kaltenbrunner for useful discussions.

## References

- [1] H. B. Nielsen, M. Ninomiya, Nucl. Phys. B 185 (1981) 20, Nucl. Phys. B 193 (1981) 173, Phys. Lett. B 105 (1981) 219.
- [2] R. Narayanan, H. Neuberger, Nucl. Phys. B 412 (1994) 574, Nucl. Phys. B 443 (1995) 305; H. Neuberger, Phys. Lett. B 417 (1998) 141.
- [3] P. H. Ginsparg, K. G. Wilson, Phys. Rev. D25 (1982) 2649.
- [4] H. Neuberger, Phys. Rev. Lett. 81 (1998) 4060.
- [5] J. van den Eshof, A. Frommer, T. Lippert, K. Schilling, H. A. van der Vorst, Comput. Phys. Commun. 146 (2002) 203.
- [6] For a recent review, see M. Stephanov, Proc. of Science LAT2006 (2006) 024.
- [7] J. Bloch, T. Wettig, Phys. Rev. Lett. 97 (2006) 012003.
- [8] D. B. Kaplan, Phys. Lett. B288 (1992) 342.
- [9] Y. Shamir, Nucl. Phys. B406 (1993) 90.
- [10] V. Furman, Y. Shamir, Nucl. Phys. B439 (1995) 54.
- [11] P. Hasenfratz, F. Karsch, Phys. Lett. B125 (1983) 308, Phys. Rept. 103 (1984) 219.
- [12] J. B. Kogut, et al., Nucl. Phys. B225 (1983) 93.
- [13] N. Dunford, J. Schwartz, Linear Operators, Part I: General Theory, Interscience Publishers, 1958.
- [14] G. H. Golub, C. F. Van Loan, Matrix Computations, The Johns Hopkins University Press, 1989.
- [15] E. D. Denman, A. N. Beavers, Appl. Math. Comput. 2 (1976) 63.
- [16] J. Roberts, Internat. J. Control 32 (1980) 677.
- [17] C. Kenney, A. Laub, SIAM J. Matrix Anal. Appl. 12 (1991) 273.
- [18] N. Higham, Linear Algebra and its application 212/213 (1994) 3.
- [19] R. A. Horn, C. R. Johnson, Topics in matrix analysis, Cambridge University Press, Cambridge, 1994.
- [20] J. Bloch, T. Wettig, arXiv:0709.4630 [hep-lat].
- [21] A. Boriçi, Phys. Lett. B 453 (1999) 46, J. Comput. Phys. 162 (2000) 123.
- [22] C. Kenney, A. Laub, IEEE Trans. Autom. Control 40 (1995) 1330.
- [23] N. J. Higham, Num. Algorithms 15 (1997) 227.
- [24] <http://www.caam.rice.edu/software/ARPACK>.
- [25] H. A. van der Vorst, J. Comput. Appl. Math. 18 (1987) 249.
- [26] V. Druskin, A. Greenbaum, L. Knizhnerman, SIAM J. Sci. Comput. 19 (1998) 38.

- [27] E. Gallopoulos, Y. Saad, On the parallel solution of parabolic equations, in: R. D. Groot (Ed.), Proceedings of the International Conference on Supercomputing 1989, Heraklion, Crete, June 5-9, 1989, ACM press, 1989.
- [28] M. Hochbruck, M. E. Hochstenbach, Subspace extraction for matrix functions, preprint (2005).
- [29] Y. Saad, SIAM Journal on Numerical Analysis 29 (1992) 209.
- [30] Y. Saad, SIAM J. Matrix Anal. Appl. 18 (1997) 435.
- [31] R. Morgan, W. Wilcox, Deflated iterative methods for linear equations with multiple right-hand sides, Tech. rep., Baylor University (2004).
- [32] A. Hasenfratz, P. Hasenfratz, F. Niedermayer, Phys. Rev. D72 (2005) 114508.
- [33] <http://www.netlib.org/lapack>.
- [34] <http://math-atlas.sourceforge.net>.
- [35] R. H. Bartels, G. W. Stewart, Commun. ACM 15 (1972) 820.
- [36] G. H. Golub, S. Nash, C. F. Van Loan, IEEE Trans. Autom. Control 24 (1979) 909.
- [37] E. S. Quintana-Ortí, R. A. van de Geijn, ACM Trans. Math. Softw. 29 (2) (2003) 218.